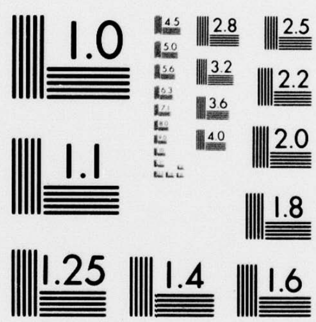
AD-A074 018 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
CAPACITY ALLOCATION TECHNIQUES IN A MULTI-USER SATELLITE SYSTEM--ETC(U)
JUL 79 J I CAPETANAKIS
UNCLASSIFIED TN-1979-34 ESD-TR-79-185 NL

| OF |
AD
A074018

END
DATE
FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

CAPACITY ALLOCATION TECHNIQUES IN A MULTI-USER SATELLITE SYSTEM

J. I. CAPETANAKIS

Group 64

TECHNICAL NOTE 1979-34

18 JULY 1979

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

ABSTRACT

The multiple accessing of a satellite by a large number of independent users is considered. The messages are generated randomly, and they are of fixed length. Depending on the accessing protocol, such a system may exhibit congestion instabilities, long message delays, or low throughput.

Several important protocols (that are applicable to such a system and which fall into the categories of direct access, reservation access, and polling access) are presented and evaluated. The E{delay} vs E{throughput} performances of the various protocols are obtained and compared. It is shown (for a specific example) that demand assignment access with a TREE reservation protocol is best for throughput up to about 70%, whereas polling access is best in heavier message traffic.

CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

This report considers the time domain multiple accessing of a satellite channel by a set of spacially isolated sources. The spacial isolation property restricts the means of communication among the sources to be only through the satellite channel. Consequently, since the requests for the channel must also be transmitted through the channel itself, problems may arise in allocating channel capacity. These problems may manifest themselves as long message delays, as low throughput, and as system instability. These three issues are defined precisely below. Loosely, however, delay corresponds to the queueing and transmission times, throughput corresponds to the channel utilization, and stability refers to the possibility that the channel becomes so congested that many users are attempting to access the channel but very few are succeeding.

Several stable multiple accessing protocols are studied here. They are classified into three categories: direct access (DA), reservation access (RA), and polling access (PA). In DA, the message is directly transmitted and message contention either is avoided by dedicating (in advance) portions of the channel capacity to each source (as in the synchronous time division multiple access (TDMA)) or is allowed to occur and then resolved with conflict resolving algorithms such as the TREE algorithm (see Section II-B or references [1,2] for a description of the TREE) or the Aloha protocol [3].

In the RA protocols, a message may be transmitted only after capacity is reserved for it. Reservations are accomplished by transmitting reservation request packets (usually much shorter than the messages) over a portion of the uplink channel. Two reservation channel protocols are examined in this report, TDMA and the TREE algorithms. Notice that the multi-accessing issues that are of interest to the whole channel in DA become significant in the reservation channel in RA.

In PA, each source is polled in sequence to determine if it has any messages to transmit. When polled, a source will transmit all the messages in its buffer before giving up the channel. The channel hand-over is accomplished with an "end of message" statement (EMS) which must be transmitted whether or not the source has messages to transmit. That is, an EMS, besides indicating the end of a message, may also indicate that there are no messages to transmit.

All the multi-accessing techniques considered here (with the exception of DA-TDMA) require a processor along with a feedback channel. The processor may be centralized (e.g., at the satellite) or distributed (i.e., each source executes the algorithm). The feedback channel is necessary so that (depending upon the multi-accessing technique used) the channel state may be determined, reservation requests may be acknowledged, or polling may be carried out. Notice that in the case of distributed control, the feedback channel may simply be a broadcast retransmission of the uplink. Since the types of processing and feedback channels used depend on such diverse factors as type of satellite (i.e., processing or not) and the availability of a reliable broadcast channel and since, in addition, these aspects are not central to the main issues of multi-accessing (i.e., delay, throughput, and stability), details of their design will not be explicitly addressed in this report.

A. System Model

The system of interest here has $N=2^n$ sources. All data messages have length s, whereas reservation requests and EMS's have length r.* The message arrival process at each source is Poisson with average arrival rate λ messages/message length/source. Each source has infinite buffer capacity so that messages which are not successfully transmitted are queued for transmission or retransmission at a later time. The buffers are allowed to be

*Often uplink modulation techniques require either time/frequency precorrection or the use of a preamble. In PA with preambles, an EMS when a message is not present may be longer than an EMS when a message is present. This is because in the latter case the same preamble may be used for the message and the EMS. Here, however, as indicated above, the length of the EMS will be taken to be r in both cases. Consequently, the delay results presented here may be considered to be upper bounds to the case in which the EMS length is less than r.

infinitely long for mathematical convenience. As will be seen, the actual buffer size need not be very large. One advantage, of the infinitely long buffer is that if the E{delay} is finite, then E{throughput} = E{arrival rate}.

It is also assumed, that the uplink consists of U identical orthogonal (in frequency) channels, R of which are used for reservation in RA, and that the channel time is slotted (i.e., synchronized). The reservation channel slots and data channel slots are r and s seconds long, respectively, and a data packet or a reservation packet is transmitted within only one slot. Although messages that are longer than one slot are not explicitly considered here, they may be processed by transmitting them over several slots. Finally, in order to emphasize the multi-accessing issues of the system, the channels are assumed to be noise free.

B. The Issues

As has been indicated above, the three main issues that are of interest in the evaluation of a multi-access system are average delay, average throughput, and system stability. Precise definitions follow:

Average delay: the total delay of all the messages in a large interval divided by the number of messages in that interval. (The message delay is the time from the instant the message is generated to the instant it is successfully received.) Here, δ will designate the delay.

Average throughput: the ratio of the number of packets transmitted over a very long interval to the number that could have been transmitted with continuous transmission. Here, h will designate the system throughput per uplink channel.

Stability: the system is stable if there exists a range in the arrival rate in which the average delay is finite.

This report emphasizes the $E\{\delta\}$ vs $E\{h\}$ performance. A by-product of the E{delay} analysis is the determination that the considered protocols are stable. The $E\{\delta\}$ is expressed in message lengths, and the $E\{h\}$ is expressed in successful message transmissions per message slot per unplink.

C. Outline and Summary

The remainder of this report is organized into 4 sections. Section II considers direct access systems. The E{delay} vs E{throughput} performance is obtained for TDMA, the static binary TREE, and the dynamic TREE protocols. These results are illustrated in Fig. 2.1 for N=1024 and U=8.

In Section III, the reservation access systems are considered. The reservation protocols that are analyzed are TDMA and the dynamic TREE. The delay results are illustrated in Fig. 3.1 for N=1024, U=8, r/s=.2, and R=1,2,4. In addition, resource pooling of the data channels is considered. The effectiveness of resource pooling is demonstrated in Fig. 3.2.

Section IV considers polling access systems. The delay results are illustrated in Fig. 4.1 for N=1024 and r/s=.2. Section V is the conclusion. For comparison, the E{delay} results of all the protocols, which are considered in this report, are illustrated in Fig. 5.1. Notice from this figure that all the protocols are stable.

The delay results for TDMA protocols have been obtained analytically, whereas the delay results of the TREE protocols were obtained by simulation. Consequently, an appendix is included which presents a derivation of the E{delay} of TDMA.

II. DIRECT ACCESS SYSTEMS

In this section TDMA, the static binary tree and the optimum dynamic tree protocols are introduced and analyzed. The objective of the analysis is the determination of E{delay} vs E{throughput} performance. The results are illustrated in Fig. 2.1.

A. Direct Access with TDMA (DA-TDMA)

In TDMA, if F sources are assigned to the channel, then each source is allocated an s-second slot every F slots. A source may transmit a message only in its designated slot. Note that if a user does not have a message to transmit, the slot will not be used; whereas if it has multiple messages to transmit, it will transmit them over several F-slot frames. Since there are U orthogonal uplink channels, the N sources are divided into U equal groups, and each group is assigned one of the channels so that $F=N/U$. From symmetry, one notes that the $E\{\delta\}$ vs $E\{h\}$ performances of the U groups are identical. In addition, since all the messages that are generated by a source are eventually transmitted in a stable system, it follows that $E\{h\} = \mu$ for $E\{\delta\} < \infty$, where $\mu = N\lambda/U$ is the aggregate arrival rate per uplink channel. In the following, $E\{\delta\}$ vs μ is obtained.

In TDMA, δ may be considered to be the sum of W, the time spent in the buffer (i.e., the time from the instant a message arrives to the instant it is successfully transmitted), and τ, the round trip delay (τ includes the delay in passing through the satellite). If δ, W, and τ are expressed in units of message length, the overall delay may be written as

$$\delta = W + \tau \quad . \tag{2.1}$$

Since τ is deterministic,

$$E\{\delta\} = E\{W\} + \tau \quad . \tag{2.2}$$

The quantity τ is independent of μ, and usually it is given. $E\{W\}$, on the other hand, depends on μ, and its determination is greatly facilitated by the recognition that a TDMA system is very similar to the classic MD1 queue. (MD1 = Markovian arrivals, deterministic service time, and a single server.) The server here is the channel, the customer is the message and the service time is sF secs. There, however, is a subtle difference between MD1 and TDMA. Whereas in MD1 a busy period is initiated immediately upon arrival of a new customer, in TDMA a busy period may begin only at multiple intervals of sF. The average delay for TDMA is derived in the appendix, and it is repeated below.

$$E\{\delta\} = 1 + \frac{F}{2} + \frac{\mu F}{2(1-\mu)} + \tau \text{ (in message lengths).} \tag{2.3}$$

Equation (2.3) is illustrated in Fig. 2.1 for F=128, where F=N/U, N=1024, τ=0, and U=8. (Normally τ is much smaller than the other terms of (2.3).) Thus, setting τ=0 is a reasonable approximation. However, if the value of τ is significant, Fig. 2.1 can easily be changed to take nonzero values of τ into consideration.) From Fig. 2.1, as well as from (2.3), it is evident that the minimum E{delay} of DA-TDMA is 1 + F/2 and that the maximum E{throughput} is one packet/slot/channel. Notice, however, that as the channel utilization approaches one the average delay approaches infinity.

For simplicity, it has been assumed that the size of the message buffer is infinite. In practice, however, infinite size buffers are not possible. One measure of the required buffer is $E\{q\}$, the average queue length per source. $E\{q\}$ is obtained by applying Little's formula to the average time spent in the buffer. That is,

$$E\{q\} = \lambda(E\{\delta\}-\tau) \quad . \tag{2.4}$$

From (2.3) and (2.4) and the definition of λ, $E\{q\}$ in message units may be written as follows:

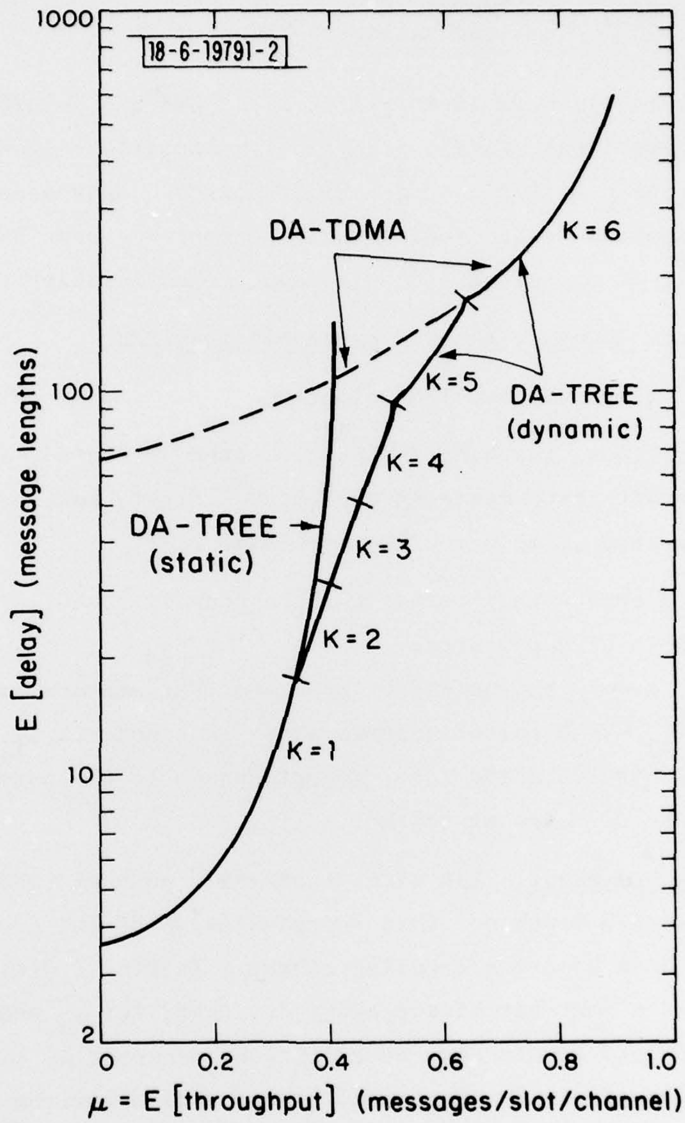


Fig. 2.1. E{delay} vs E{throughput} for direct access systems (1024 users and 8 uplink channels).

$$E\{q\} = \frac{\mu}{2} + \frac{\mu U}{N} + \frac{\mu^2}{2(1-\mu)} \quad . \tag{2.5}$$

Equation (2.5) is illustrated in Fig. 2.2 for U=8 and N=1024. As can be seen, the average queue length is acceptable for reasonable values of throughput. For example, $E\{q\} \leq 2$ for $\mu < .8$. $E\{q\}$, however, approaches infinity quite rapidly as μ approaches 1. Notice that the system cannot be efficiently operated near $\mu=1$, since the $E\{\delta\}$ also approaches infinity in this region.

B. Direct Access With TREE Protocols (DA-TREE)

1. Static Binary TREE Algorithm

In this section, the binary TREE algorithm is stated, an example is presented, and its performance is evaluated. First, however, some definitions concerning the tree graph are given (see Fig. 2.3).

Depth of a node - the tier at which a node is found. The root node is at depth zero.

Degree of node - the number of branches that emanate from a node.

Subtree T_{ij} - the rooted subtree whose root node is n_{ij}, where, in a binary rooted tree, j corresponds to the particular one of the 2^i nodes at depth i.

Algorithm Statement: Let each of the $N=2^n$ sources correspond to a leaf on a binary tree of depth n. This representation of the sources may be considered to be a binary addressing scheme. In Fig. 2.3(a), for example, each source has a four-bit binary address. Also, let T_x and T_y be two binary rooted subtrees and assume that no collisions occurred up to the beginning of the present pair of slots. Then, the binary tree algorithm is as follows:

1. Choose $T_x = T_{10}$; $T_y = T_{11}$.
2. Transmit all the packets in T_x in the first slot of the present pair of slots, and transmit all the packets in T_y in the second slot. (Each source may process at most one packet per conflict resolving interval.)

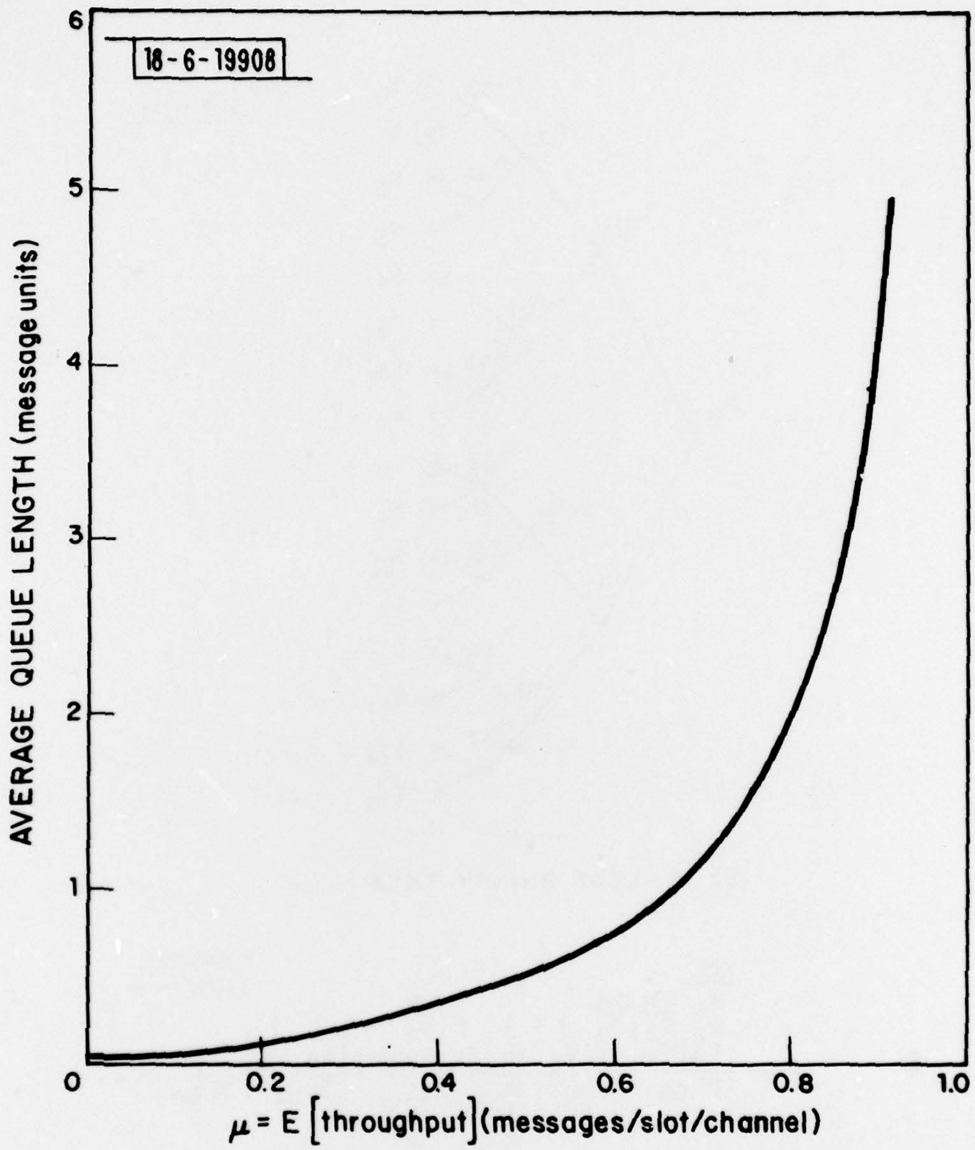
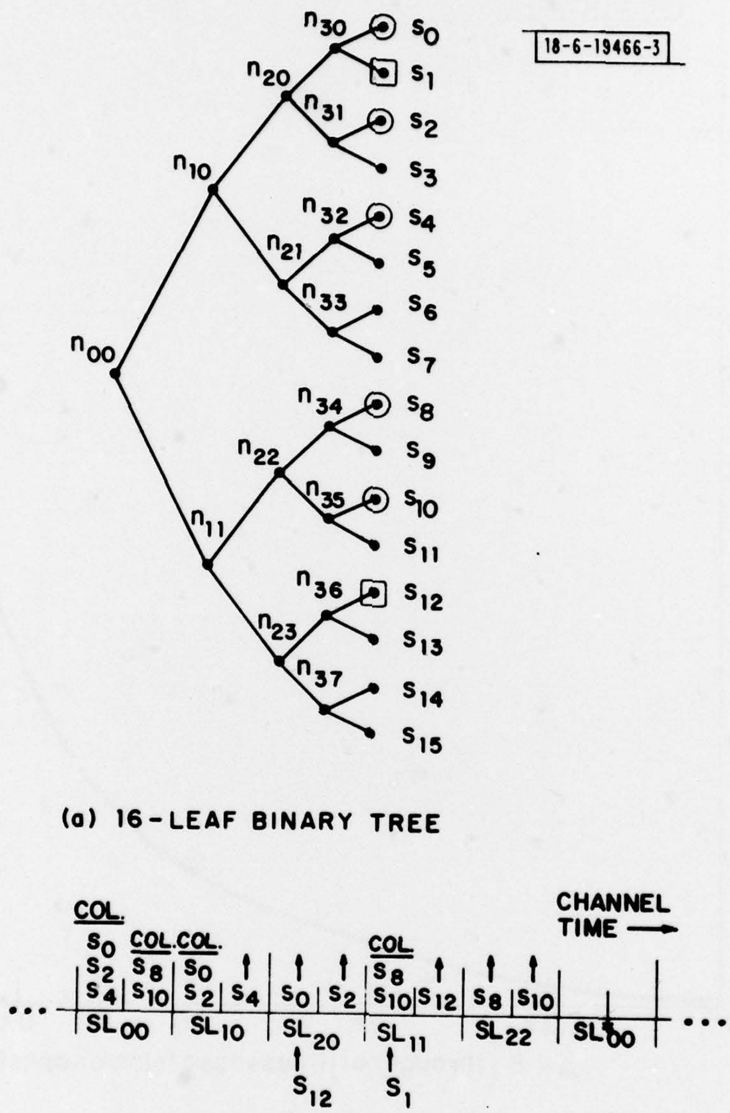


Fig. 2.2. E{queue length} vs E{throughput} for TDMA (1024 users and 8 uplink channels).



(b) SLOTTED SATELLITE TIME

Fig. 2.3. An example of the binary TREE protocol.

3. If any collisions occur in the preceding step, then

   a. Until these collisions are resolved, no new packets are transmitted unless they are generated by a source belonging to branch which is known to have an unresolved collision.

   b. Resolve the first collision (if any) before resolving the second (if any).

   A collision in T_x (or T_y) is resolved by dividing T_x (or T_y) into two halves (say A and B); setting $T_x = A$, $T_y = B$; and then repeating steps 2 and 3.

Example: Let there be 16 sources $\{S_0, S_1, \ldots, S_{15}\}$ and let each correspond to a leaf on the 16-leaf binary tree as shown in Fig. 2.3(a). Figure 2.3(b) depicts the slotted satellite time. Note that the slots are paired and that a slot pair is designated by SL_{ij}. For convenience, the round-trip delay τ is taken as zero. (If $\tau > 0$, then the slot pairs should be separated by at least τ. As is shown below, the issue of a non-zero τ may be efficiently overcome by interleaving two algorithms so that the results of one transmission are received before it is time to retransmit (see Fig. 2.4).)

Now assume that no collisions have occurred until the beginning of SL_{00}, when sources S_0, S_2, S_4, S_8 and S_{10} each has a packet to transmit. Then beginning with SL_{00} where the first contention arises, the TREE algorithm takes the following steps in the indicated slot pairs:

SL_{00} All the sources in T_{10} that have packets to transmit (i.e., S_0, S_2 and S_4) do so in the first slot of SL_{00}, and the corresponding sources in T_{11} do so in the second slot. This results in two collisions, one among S_0, S_2, and S_4 and the other between S_8 and S_{10}. Since there was at least one collision in SL_{00}, most* new packets that arrive are not transmitted until that collision is resolved.

---

*As previously mentioned, the exception involves new packets which are generated by a source in a branch with an unresolved conflict.

SL_{10}	Since there was a collision in T_{10}, the sources at T_{10} are divided in half and the packets in T_{20} and T_{21} are transmitted in the first and second slots of SL_{10} respectively. This results in a collision between S_0 and S_2 and in a successful transmission by S_4.
SL_{20}(a)	S_{12} receives a new packet (which can be processed during the subsequent efforts to resolve the conflict in T_{11}).
SL_{20}(b)	Since there was a collision in T_{20}, T_{30} and T_{31} transmit their packets in the first and second slots of SL_{20}, respectively. This results in two successful transmissions by S_0 and S_2.
SL_{11}(a)	S_1 receives a new packet. (This cannot be processed until the original collision is resolved.)
SL_{11}(b)	Since there was a collision in T_{11}, T_{22} and T_{23} transmit their packets in succession. This results in a collision between S_8 and S_{10} in the first slot. In the second slot, T_{23} transmits resulting in the successful transmission by S_{12}.
SL_{22}	Since there was a collision in T_{22}, T_{34} and T_{35} transmit their packets in successions. This results in two successful transmissions by S_8 and S_{10}.
SL^*_{00}	Since each of the sources participated in at least one collision-free transmission we know that the original contention has been resolved. Any new packets that may have arrived (and not been processed) to T_{10} during this conflict resolution interval are transmitted in the first slot of SL^*_{00}, and packets that arrived to T_{11} are transmitted in the second slot. The process continues, as described above.

In this example, 10 slots were used to transmit 6 packets.

The E{delay} vs μ performance of the static binary TREE has been obtained by simulation for N=1024 and U=8. Here, as in TDMA, the sources are divided into eight equal groups, one for each uplink channel. There are 128 sources per group and each group is allowed to access only one of the eight channels.

In addition, it is assumed that the round-trip delay is greater than zero but less than two message lengths. Therefore, two TREE algorithms, each processing 64 sources, are interleaved over each channel as shown in Fig. 2.4. (In general, the number of algorithms that are interleaved is the smallest k which satisfies $\tau \leq 2(k-1)$; τ is the round-trip delay expressed in message lengths.)

If ℓ is defined to be the number of algorithm steps that occur from the instant a message is generated to the instant it is successfully transmitted (see Fig. 2.4), then $E\{\delta\}$ (in message lengths) may be written as follows:

$$E\{\delta\} = 4E\{\ell\} - \frac{1}{2} + \tau \quad . \qquad (2.6)$$

An estimate of $E\{\ell\}$ is obtained by simulation for various values of μ. The results were substituted into (2.6) to obtain $\langle\delta(\mu)\rangle$ (an estimate of $E\{\delta\}$), which is illustrated in Fig. 2.1 along with the E{delay} results of TDMA. Notice that, in light traffic, the E{delay} of DA-TREE is about 1/30 that of DA-TDMA. However, as the traffic increases (i.e., $\mu > .47$) the DA-TDMA performance becomes better than that of DA-TREE. As shown in the following section, this suggests that a hybrid system (one that uses DA-TREE in light traffic and DA-TDMA in heavy traffic) will have better overall performance.

Finally, it can be shown that, for a given μ, the E{delay} of DA-TREE is insensitive to the number of users. This is in contrast to the E{delay} of DA-TDMA which is directly proportional to the number of users.

2. Optimum Dynamic TREE Algorithm

Whereas in the static TREE algorithm, the TREE is held fixed throughout, in the dynamic TREE algorithm, the TREE is allowed to vary depending on traffic

18-6-199909

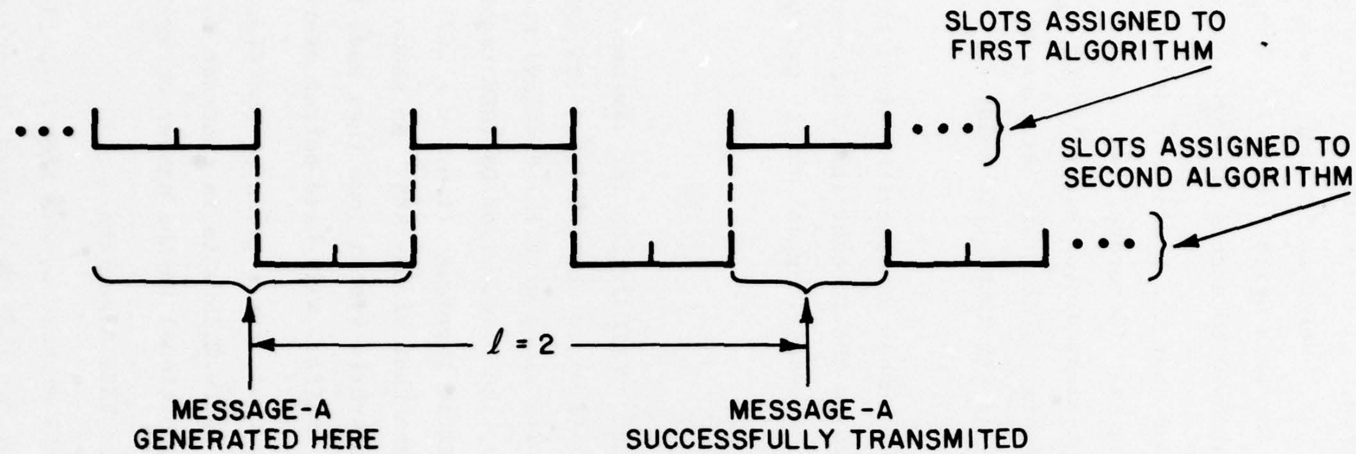


Fig. 2.4. Illustrating the interleaving of two TREE algorithms on a single channel.

conditions. The variation in the tree occurs only in the degree of the root node. That is, the tree is binary everywhere except for the root node which may have degree 2^k. Given μ, k is chosen to minimize average delay, and it is allowed to take the values $[1,2,\ldots,m]$ where 2^{m+1} is the number of sources assigned to a particular uplink.

It is of significance to note that if $k = m$, the TREE algorithm is equivalent to TDMA. That is, at $k = m$, the tree graph reduces to a single node with 2^m branches, and the protocol sequentially assigns a slot to each user. Consequently, the TREE algorithm is a generalization of TDMA, and it follows from this observation that the optimum TREE algorithm is more efficient than TDMA.

In the dynamic TREE algorithm that is considered here, for a given μ, the tree that minimizes the E{delay} is used throughout. An alternative dynamic algorithm is to vary the tree optimally as a function of not only μ but also as a function of the probabilistic distribution of the contending sources. This latter technique, which is considered elsewhere (see Ref. [1,2]), is more efficient than the one being considered here. Therefore, its performance is upperbounded by the results presented in this report.

In Fig. 2.5, $E\{\delta\}$ vs. μ, parameterized on k, is presented for $N = 1024$ and $U = 8$. Notice that as k increases, the maximum average throughput and the minimum average delay also increase. From this figure, the optimum k may be read as a function of μ. The lower envelope of the curves in Fig. 2.5 is the E{delay} vs μ performance of the optimum dynamic TREE. This envelope is also illustrated in Fig. 2.1.

3. A Suboptimum Dynamic TREE Algorithm

A somewhat simpler dynamic algorithm is the one that optimally uses either TDMA or the binary TREE. From Fig. 2.1, we see that using the binary TREE for $\mu \leq .47$ and TDMA for $\mu > .47$ minimizes the average delay. The performance of this protocol is given by the lower envelope of the binary TREE and the TDMA delay curves of Fig. 2.1. As can be seen, not much is lost over the optimum algorithm in performance, yet the gain in simplicity of implementation is considerable.

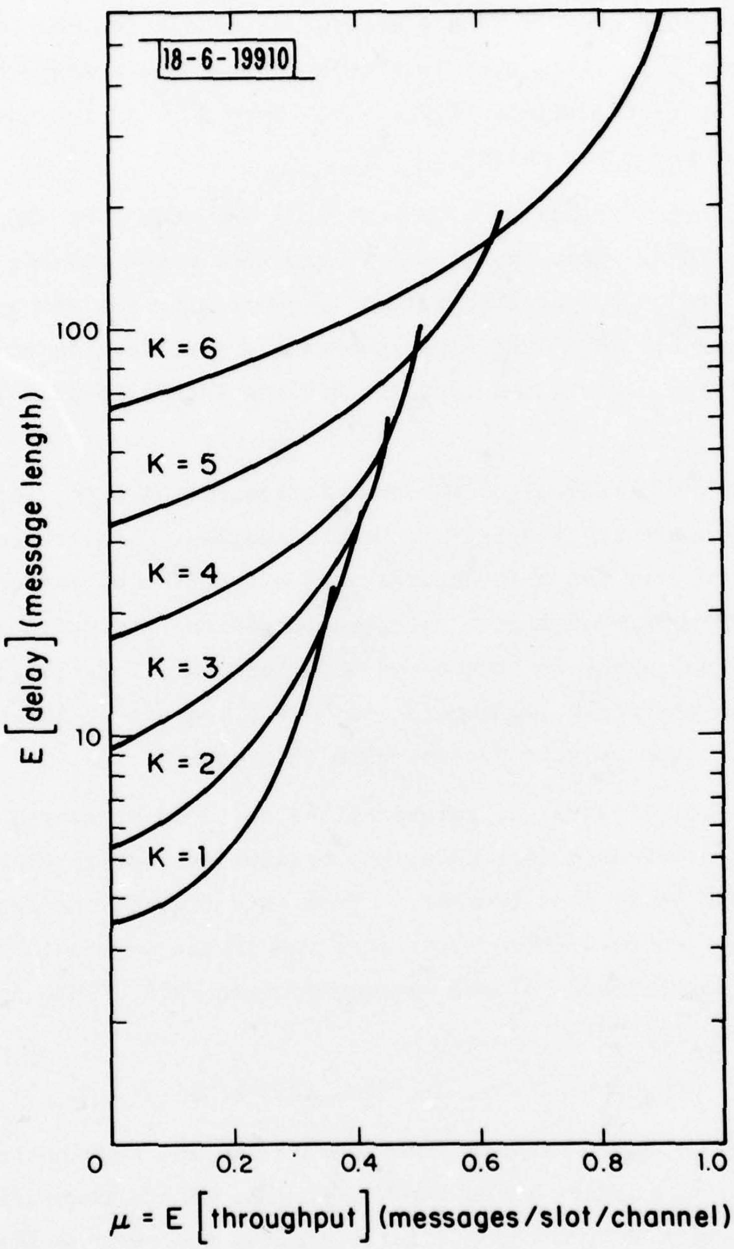


Fig. 2.5. E{delay} vs E{throughput} for direct access with the optimum dynamic TREE (1024 users and 8 uplink channels).

III. RESERVATION ACCESS SYSTEMS

In a reservation system, a message may be transmmitted only after channel capacity is reserved for that message. The reservation requests are carried out with reservation packets that are r seconds long (normally $r \ll s$). The reservation packets are transmitted over R of the U channels, while the messages are transmitted over the other U-R channels. R need not be an integer; here, however, only the integer values $R = 1,2,\ldots U-1$ are considered.

The sources need to know the status of the requests. This is usually accomplished by transmitting reservation acknowledgment packets over a feedback channel. The acknowledgment packet may vary from a simple broadcast retransmission of the request packet in a distributed processed system to a packet containing information specifying exactly when the requesting source may transmit in a centrally controlled system. Here, it will be assumed that the processing is distributed. (Since the reservation packet must be received by the processor before an acknowledgment packet can be transmitted, the delay of centralized processing is r seconds longer than that of distributed processing.)

After the reservation request is acknowledged, the message is placed on a distributed queue where it waits its turn to be transmitted. Consequently, the overall message delay may be broken up into two components D_r and D_m. D_r is the time from the instant a message is received to the instant the reservation request is acknowledged, and D_m is the time from the instant the request is acknowledged to the instant the message is successfully received at its destination. Thus,

$$E\{\delta\} = E\{D_m\} + E\{D_r\}. \tag{3.1}$$

Two assumptions are made. First, a successfully transmitted request packet will make the reservation for all the messages that are in the source buffer immediately prior to its transmission. Secondly, the message arrival process at the distributed queue is Poisson. The first assumption is natural and should be incorporated in an operational system. The second assumption results in an approximation which is quite accurate for the typical case of $r \ll s$.

Two queuing disciplines are examined for the distributed queues that are assumed here: no resource pooling (NRP) and resource pooling (RP). In NRP, N/(U-R) of the sources are assigned to each of the U-R data channels, and a source may transmit only in the channel assigned to it. Consequently, NRP may be inefficient, since it is possible for one channel to be backlogged while others may remain unused. The RP discipline alleviates this problem by allowing a message to be transmitted through any unused channel.

The TDMA and the dynamic TREE reservation protocols are considered here. The overall E{delay} for these reservation protocols is obtained for the NRP discipline. Since the distributed queue discipline does not affect D_r, the effects of the RP discipline are then examined by determining the resulting $E\{D_m\}$.

Besides the distributed queue discipline and the reservation protocol, the system performance also depends on R. Consequently, the E{delay} vs E{throughput} performance is obtained for several values of R. An interesting byproduct of this exercise is system optimization with respect to R.

A. Reservation Access with TDMA (RA-TDMA)

In this section, $E\{\delta\}$ vs μ is obtained for the reservation system that uses the TDMA protocol on the reservation channels and the NRP discipline on the data channels. The results are illustrated on Fig. 3.1.

Letting τ be the round trip delay, $E\{D_r\}$ (in units of message length) may be written as

$$E\{D_r\} = \left(\frac{N}{2R}\right) \frac{r}{s} + \frac{r}{s} + \tau \quad . \tag{3.2}$$

The first term above is the average delay until transmission, the second is the transmission time, and the third is the propagation delay.

The distributed queue delay is

$$E\{D_m\} = \frac{3}{2} + \frac{U\mu}{2(U-R-U\mu)} \qquad (3.3)$$

This follows from the assumption that the message arrival process at each uplink is Poisson with parameter $U\mu/(U-R)$ and the observation that the resulting queue is a synchronous MD1 (see the appendix). Combining (3.1), (3.2) and (3.3) results in

$$E\{\delta\} = \left(\frac{N}{2R} + 1\right)\frac{r}{s} + \frac{U\mu}{2(U-R-\mu U)} + \frac{3}{2} + \tau \quad . \qquad (3.4)$$

This is illustrated in Fig. 3.1 for $\tau = 0$, $U = 8$, $N = 1024$, $r/s = .2$ and $R = 1$, 2, and 4. The vertical asymptotes of the RA-TDMA delay curves in Fig. 3.1 are a function of the number of data channels, whereas the horizontal asymptotes are a function of the reservation channel. This observation allows us to conclude that the maximum average throughput is limited by the number of data channels, and the minimum average delay is limited by the number of reservation channels. An interesting interpretation of this is that increasing R in a system is equivalent to trading throughput for delay.

From Fig. 3.1 we also see that given μ, an R exists which minimizes E{delay}. This observation suggests a dynamic TDMA scheme in which R varies optimally as a function μ. The optimum R, for this example, may be read directly from Fig. 3.1. In general, it may be determined from (3.4). Varying R optimally, as a function of μ, results in a system whose performance is the lower envelope of the RA-TDMA curves shown in Fig. 3.1.

B. <u>Reservation Access with the Dynamic TREE Protocol (RA-TREE)</u>

The reservation protocol that is considered here is the optimum dynamic TREE. The overall delay for the RA-TREE may be decomposed, as in RA-TDMA, into D_m and D_r. $E\{D_m\}$ is the same for both schemes and it is given by (3.3). $E\{D_r\}$, on the other hand, depends on the specific reservation protocol. The tree graph for the optimum dynamic TREE protocol is binary everywhere except for the root node whose degree may be 2^k, $k = 1,2,\ldots \frac{N}{2R}$. The root node degree

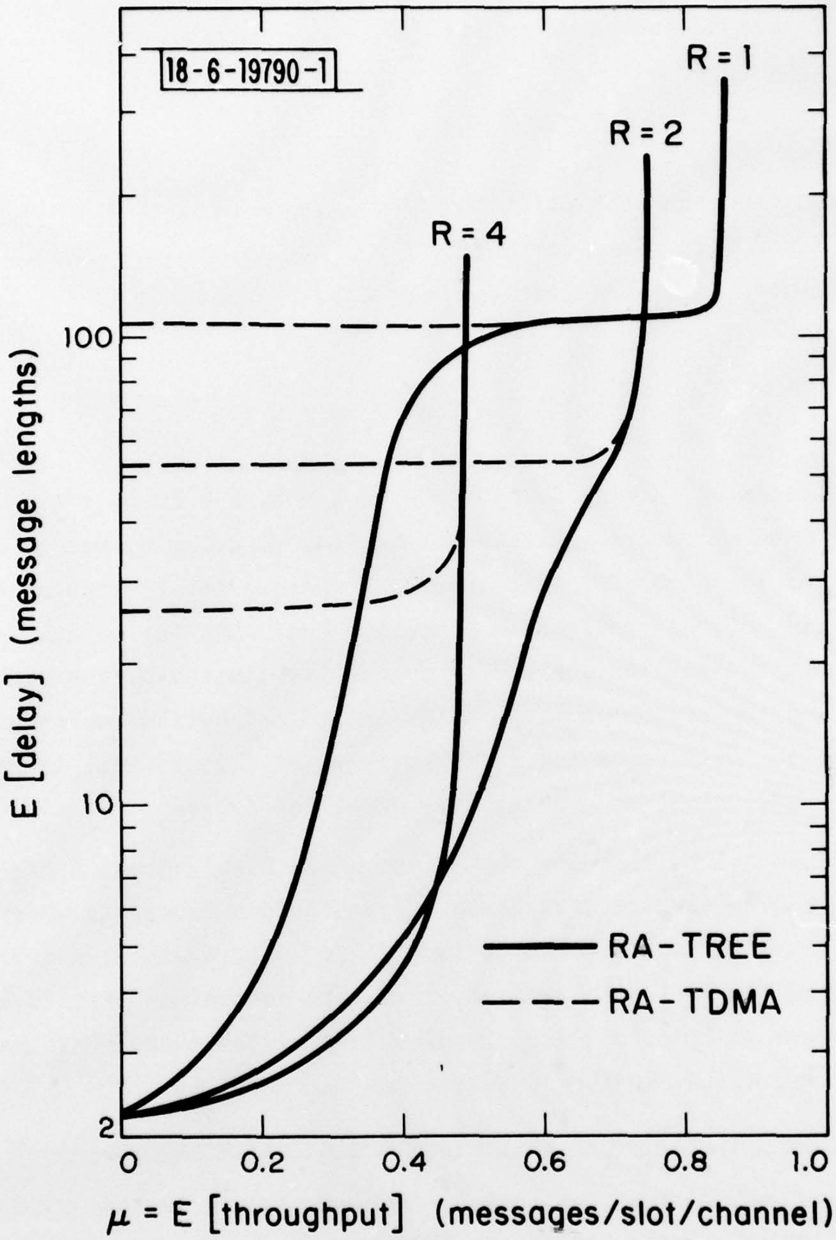


Fig. 3.1. E{delay} vs E{throughput} for reservation access systems (1024 users, 8 uplink channels, and message length to reservation packet length ratio equal to 5).

is chosen to minimize $E\{D_r\}$.

Due to the nonzero round trip delay associated with actual satellite communications systems, two TREE algorithms are interleaved per uplink channel as shown in Fig. 2.4. (If $\tau > 2r$, it may be more efficient to interleave more than two algorithms). The resulting $E\{D_r\}$, in terms of $E\{\ell\}$, is given in message units by

$$E\{D_r\} = [4E\{\ell(\mu,N/R)\} - 1/2]\ \frac{r}{s} + \tau \quad . \tag{3.5}$$

Where, N/R in the argument of ℓ is the number of sources per reservation channel. $E\{\ell(\mu,\ N/R)\}$ was simulated for $N = 1024$ and $R = 1,2,4$. This simulation is similar to that of DA-TREE except for one point: whereas in RA-TREE, one successful transmission of a reservation request effectively empties the source buffer, in DA-TREE, each data packet individually must contend for the channel. The results of the simulation were combined with (3.1), (3.3), and (3.5) to obtain the $E\{\delta\}$ that is illustrated in Fig. 3.1. Notice from this figure that, for any given R, RA-TREE is significantly more efficient than RA-TDMA. Also note that, the system may be optimized with respect to R. That is the lower envelope to these curves is the best that can be attained by varying R optimally with respect to μ.

C. Resource Pooling

As has been indicated resource (channel) pooling effects only the D_m component of the delay (i.e., the delay of the distributed queue). Thus, $E\{D_m\}$ vs μ has been obtained by simulation for $N = 1024$, $U = 8$, and several values of R. The results are illustrated in Fig. 3.2, where for comparison the corresponding $E\{D_m\}$ for NRP is also shown. As expected, RP does improve the delay performance. This improvement can be substantial, especially if the number of data channels (i.e., U-R) being pooled is large.

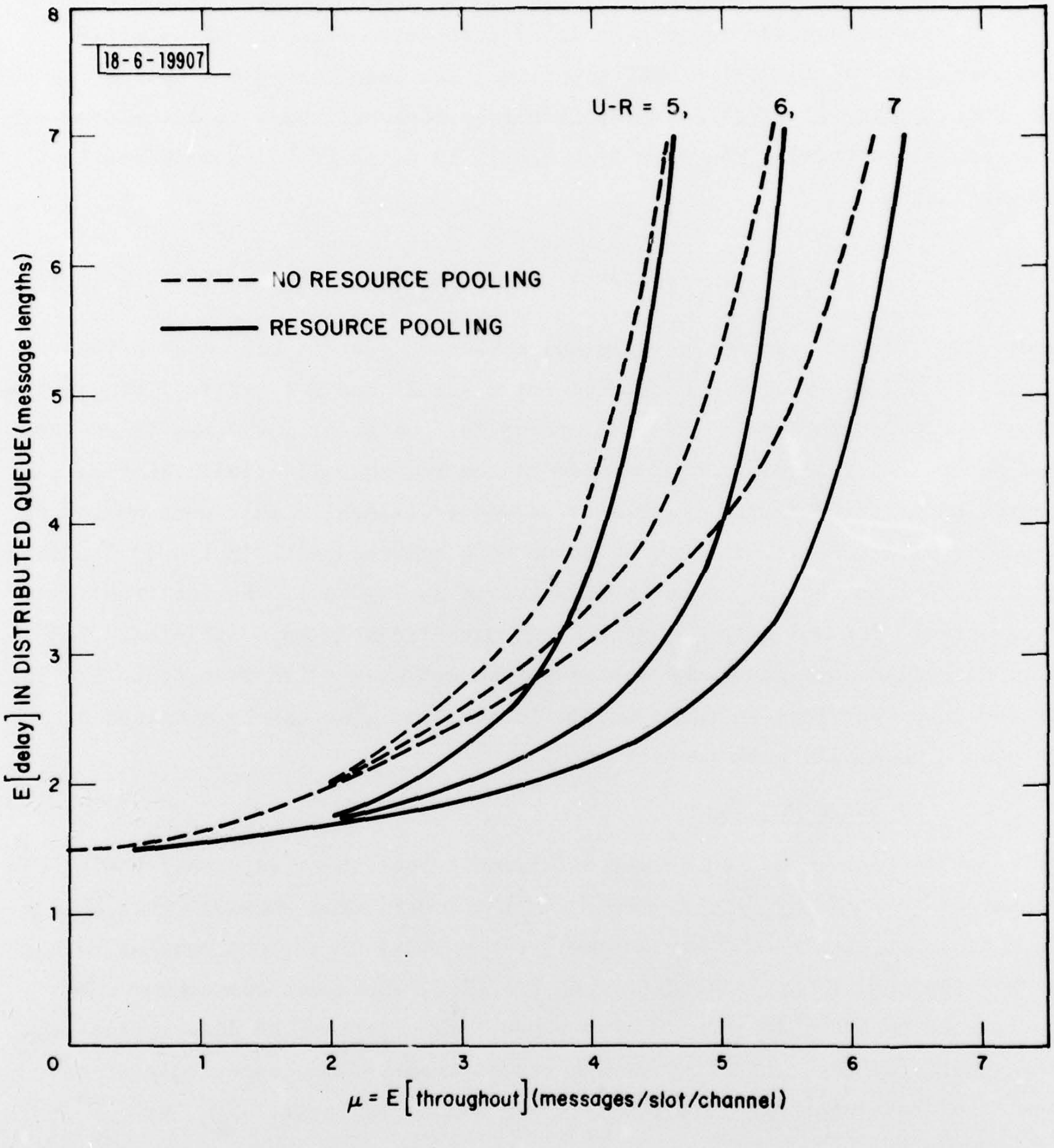


Fig. 3.2. Illustrating the effects of resource pooling (1024 users).

IV. POLLING ACCESS SYSTEMS

In PA the sources are polled in a predetermined order. When polled, a user transmits all the messages that may be in its buffer. At the end of a message transmission, a source hands over the channel to the next source by transmitting an end of message statement (EMS). If there are no messages to transmit, the user is still required to transmit an EMS. In such cases, the EMS serves the channel hand-over function.

The polling may be carried out centrally by a central processor or distributedly by each source. In the former case the EMS is transmitted to the central processor. After receipt of the EMS, the processor informs the next source on the list that it may begin transmission. If the polling is carried out distributedly, the EMS is transmitted directly (perhaps in a broadcast manner) to the next source on the polling list. Distributed polling may be a little faster since the EMS and the polling statement are the same thing. However, the robustness of the distributed system should be considered, since it is possible for the whole system to become inoperative if a single source fails to pass-on the end of message statement.

Polling Access has been analyzed in [5], where the following expression for $E\{\delta\}$ is derived:

$$E\{\delta\} = \frac{1}{2}\,\frac{\mu}{1-\mu} + \frac{1}{2}\,\frac{r_o(\frac{N}{U} - \mu)}{1-\mu} + \frac{1}{2}\,(1 - \frac{\mu U}{N}) + (1 + \tau) \quad . \qquad (4.1)$$

The quantity r_o in (4.1) is the time between the end of the message transmission by one source and the beginning of the transmission by another (i.e., it includes the end of message statement, the polling statement and the propagation delay of these statements). The quantity $(1+\tau)$ in (4.1), which is not included in the expression of $E\{\delta\}$ given in [5], corresponds to the transmission and propagation delay of one message unit.

Equation (4.1) is illustrated in Fig. 4.1 for $\tau = 0$, $r_o = s/r = .2$, $U = 8$, and $N = 1024$. Notice that the E{delay} performance of PA is considerably better than that of it close relative RA-TDMA.

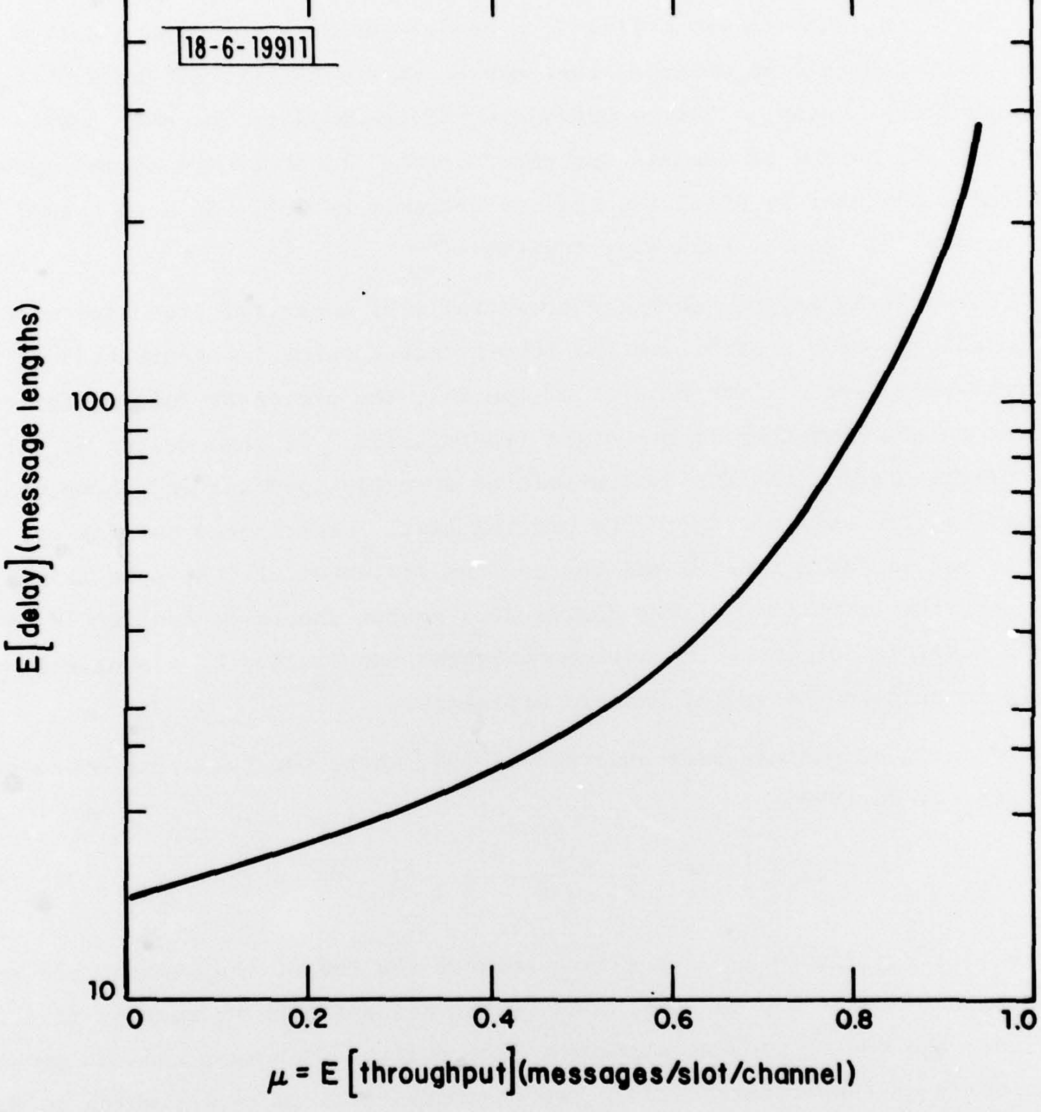


Fig. 4.1. E{delay} vs E{throughput} for polling access systems (1024 users and 8 uplink channels).

V. CONCLUSIONS

The following time domain multi-accessing protocols have been considered: direct access TDMA, direct access with TREE (static as well as dynamic), reservation access with TDMA, reservation access with the optimum dynamic TREE, and polling access. These protocols have been presented, and their E{delay} vs E{throughput} performances have been determined and evaluated for a system of 1024 users, 8 orthogonal uplinks, and message length to reservation packet length ratio of 5.

For comparison, the E{delay} performances of all the protocols are illustrated together in Fig. 5.1. DA-TDMA, being the easiest scheme to implement may be considered to be the baseline design. Its delay property, however, is relatively poor, especially in light message traffic. It is evident from Fig. 5.1 that (for this example) RA-TREE is superior for E{throughput} <.70 packets/slot/channel, whereas PA is superior for E{throughput} >.70. [RA-TREE would have been superior throughout had it been optimized over all real R.]

The observation about the relative performances of RA-TREE and PA suggests that the most efficient scheme (in terms of average delay) is a hybrid between the two. The performance of other hybrid schemes may be obtained from Fig. 5.1 simply by taking the lower envelope of the appropriate curves.

In principle, PA and RA-TDMA are very similar. In PA the controller asks the users if they have messages to transmit, whereas in RA-TDMA the users ask (or request) the control for channel capacity. In addition, they both require the transmission of a short message (i.e., request packet or end of message statement), and they both require a control and a feedback channel. However, as can be seen from Figs. 3.1 and 4.1 (or from Fig. 5.1), PA is the more efficient of the two - especially if R is held fixed. There is one subtle advantage to RA-TDMA over PA, however. Whereas in RA-TDMA, a user is guaranteed access to the control within a fixed time (i.e., one cycle of the reservation slots), in PA the maximum time to access the control increases as the traffic increases. For most applications, however, this disadvantage is insignificant compared to the average throughput and delay efficiencies of PA.

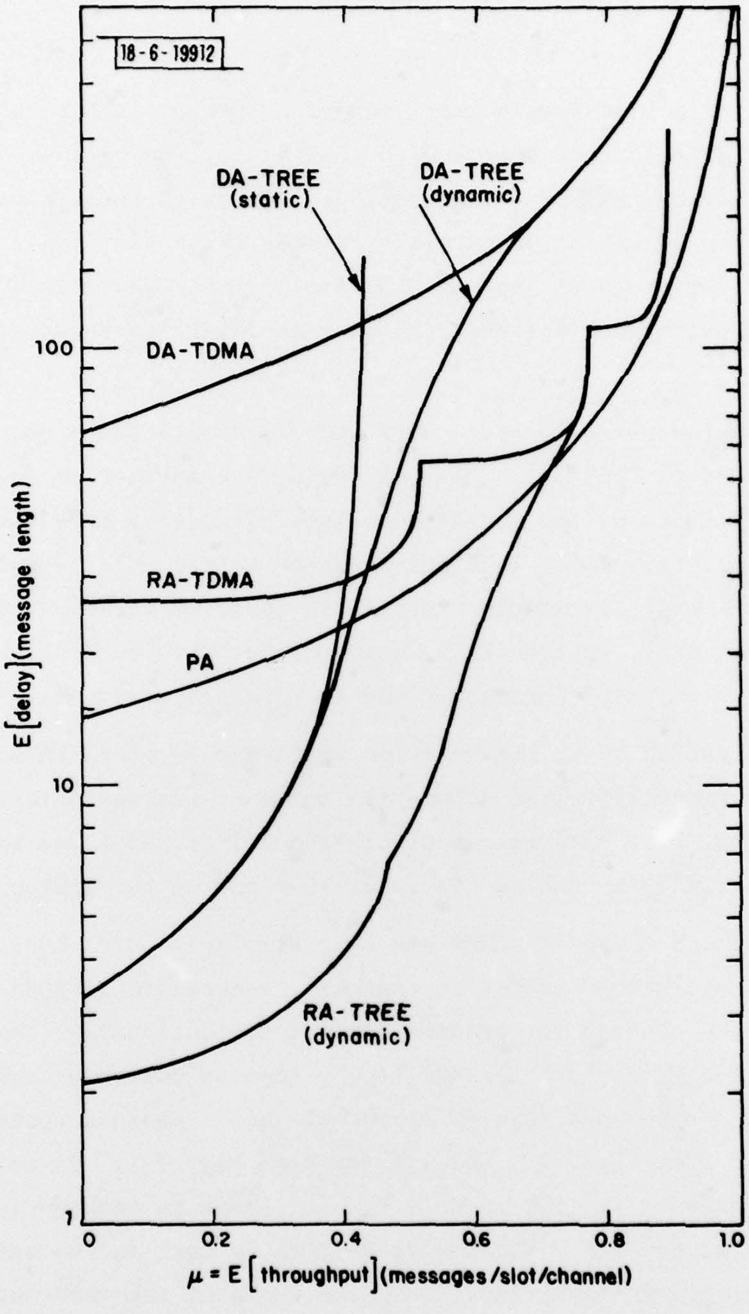


Fig. 5.1. Comparing the E{delay} performances of the varous potocols (1024 users and 8 uplink channels).

This report has also considered the effectiveness of resource pooling in reservation access systems. (In resource pooling, a user may use any available channel to transmit a message rather than being restricted to a specific channel.) The average delay under pooling and no pooling has been obtained. The results are illustrated in Fig. 3.2. As can be seen, when the number of channels is large resource pooling is considerably more efficient.

To facilitate the analysis the terminal message buffers were assumed to have infinite capacity. In order to obtain a measure on the actual required buffer length, the E{queue length} was obtained for DA-TDMA (see Fig. 2.2). The result is that for TDMA, in the operating range of interest (i.e., less than 80% channel utilization), the average queue length is not large (i.e., less than message units). The average queue length for the other protocols can be derived easily by applying Little's formula to the corresponding E{delay} results.

Although all the results (with the exception of those for DA-TREE and RA-TREE) have been derived for a general system of N users, U uplinks, r/s reservation to data slot length ratio, and τ round trip delay, they have been evaluated, illustrated, and compared for N = 1024, U=8 and τ=0. An obvious question, therefore, is how do the various protocols compare under a different set of parameters. In general, for DA-TDMA, RA-TDMA, and PA, the E{delay} is an increasing function of N/U (see (2.3), (3.4) and (4.1). The E{delay} of DA-TREE and RA-TREE, on the other hand, can be shown to be insensitive to N/U. Consequently, increasing N/U will deteriorate the performance of DA-TDMA, RA-TDMA and PA relative to DA-TREE and RA-TREE.

The quantity r/s is of interest in comparing reservations and polling access systems to direct access systems. In general, as r/s decreases the performances of PA and DA improve relative DA. Finally, the effects of a non-zero τ can easily be taken into consideration since τ is an additive constant to all the delay equations.

APPENDIX: THE AVERAGE MESSAGE DELAY IN TDMA SYSTEMS

Define a synchronous MD1 queue (SMD1) to be the conventional MD1 queue (see eq. (5.67) in [4]) with the added restriction that busy periods may begin only at integer multiples of the service time. In this appendix, first, a relationship is proved between the average waiting times of the conventional MD1 queue and the SMD1 queue. Secondly, this result is extended to the average message wating time in a TDMA system. By waiting time, we mean the time spent by a customer in queue plus the time spent in the service facility.

Theorem: Let D_s and D_a be the average waiting times of the SMD1 and the conventional MD1 queues, respectively, and let X be the service time. Then,

$$D_s = D_a + \frac{X}{2} \quad . \tag{A.1}$$

Proof: Choose a sample function from the Poisson arrival process and apply this particular function to an SMD1 queue and an MD1 queue. Next, choose any busy period in the MD1 queue, let $\underline{C} = \{C_i;\ i = 1, 2, \ldots k\}$ be the customers that were served in that busy period, and let $\underline{t_d} = \{t_i;\ i = 1, 2, \ldots k\}$ be the departure times of $\underline{C}$.

Now, $\underline{\hat{t}_d}$ the departure times of $\underline{C}$ in SMD1 are given by

$$\hat{t}_i = t_i + \Delta, \text{ for } i = 1, 2, \ldots k, \tag{A.2}$$

where Δ, in light of the Markovian property of the arrival process, is a random variable uniformly distributed between [0,X]. Taking expectations of both sides of (B.2) and noting that the arrival times of $\underline{C}$ in SMD1 are the same as in MD1, results in (B.1).

<u>QED</u>

Corollary: Let $D_a(X)$ be the average waiting time in an MD1 queue with service time X. Also let D_{TDMA} be the average message delay (i.e., waiting time in buffer plus transmission time of randomly chosen packet) in a TDMA system with F users each of which has infinite buffer capacity. In addition, assume that the message arrival process in Poisson, and that all message have length s. Then,

$$D_{TDMA} = D_a(Fs) + \frac{Fs}{2} -(F-1)s \qquad (A.3)$$

Proof: D_{TDMA} may be considered to be the delay of a randomly chosen customer in SDM1 queue, in which the randomly chosen customer has service time s but all others preceding it have service times Fs. Equation (B.3) follows from this observation and the preceding theorem.

<u>QED</u>

ACKNOWLEDGMENTS

The computer programming for this report was carried out by Lori L. Jeromin. Her excellent support is greatly appreciated. David McElroy's comments and suggestions have been very helpful and welcomed.

REFERENCES

1. J. I. Capetanakis, "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations," Technical Report ESL-R-806, Electronics Systems Laboratory, M.I.T. (March 1978).

2. J. I. Capetanakis, "TREE Algorithms for Packet Broadcast Channels," to be published in IEEE Trans. on Inf. Theory (September 1979).

3. L. Kleinrock and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. on Comm. COMM-23, 410 (1975).

4. L. Kleinrock, Queueing Systems Vol. I: Theory (Wiley Interscience, New York, 1975).

5. W. W. Chu and A. G. Konheim, "On the Analysis and Modeling of a Class of Computer Communications Systems," IEEE Trans. on Comm. COMM-20, 645 (1972).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) ESD-TR-79-185	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Capacity Allocation Techniques in a Multi-User Satellite System		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Note
		6. PERFORMING ORG. REPORT NUMBER Technical Note 1979-34
7. AUTHOR(s) (10) John I. Capetanakis		8. CONTRACT OR GRANT NUMBER(s) (15) F19628-78-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 63431F Project No. 2029 (26)
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Systems Command, USAF Andrews AFB Washington, DC 20331		12. REPORT DATE (11) 18 July 1979
		13. NUMBER OF PAGES 38 (12) 38p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited. (14) TN-1979-34

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

packets	data networks	satellite
demand access	protocol	queueing delay
multiple access	throughput	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The multiple accessing of a satellite by a large number of independent users is considered. The messages are generated randomly, and they are of fixed length. Depending on the accessing protocol, such a system may exhibit congestion instabilities, long message delays, or low throughput.

Several important protocols (that are applicable to such a system and which fall into the categories of direct access, reservation access, and polling access) are presented and evaluated. The E{delay} vs E{throughput} performances of the various protocols are obtained and compared. It is shown (for a specific example) that demand assignment access with a TREE reservation protocol is best for throughput up to about 70%, whereas polling access is best in heavier message traffic.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

207 650 JOB